ARL-TR-8113 ● AUG 2017

US Army Research Laboratory

Hybrid 2-D and 3-D Immersive and Interactive User Interface for Scientific Data Visualization

by Simon Su and Luis Bravo

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-8113 ● AUG 2017

US Army Research Laboratory

Hybrid 2-D and 3-D Immersive and Interactive User Interface for Scientific Data Visualization

by Simon Su
Computational and Information Sciences Directorate, ARL

by Luis Bravo
Vehicle Technology Directorate, ARL

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* August 2017	**2. REPORT TYPE** Technical Report	**3. DATES COVERED (From - To)** January 2016 – July 2016

4. TITLE AND SUBTITLE Hybrid 2-D and 3-D Immersive and Interactive User Interface for Scientific Data Visualization	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Simon Su and Luis Bravo	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory Computational and Information Sciences Directorate ATTN: RDRL-CIH-S Aberdeen Proving Ground, MD 21005-5066	**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-8113
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We present a hybrid 2-D and 3-D scientific visualization workflow that allows users to add 3-D immersive visualization capability to their day-to-day desktop scientific visualization workflows. The 3-D immersive and interactive visualization capability was added to the widely used visualization tool ParaView running on the zSpace semi-immersive virtual reality display system using ray tracing for rendering at an interactive rate. The zSpace display system supports head-tracked stereoscopic display, and stylus-based 3-D interaction. Further, the zSpace virtual reality system requires very little calibration or maintenance after the initial system driver setup and configuration. For software, we extended an existing ParaView plugin—the pvOSPRay renderer plugin—to work with the Immersive ParaView plugin to support 3-D immersive and interactive visualization. We tested the workflow presented using a data set with 8 million data points at an interactive rate. We also evaluated this visualization workflow in a pilot user study comparing 2-D visualization and 3-D immersive and interactive visualization using a scientific simulation data set common to our users. We collected performance data and user feedback from a usability questionnaire. The 3-D immersive and interactive visualization workflow was found to be preferred over the 2-D visualization workflow, together with some very constructive user feedback on the 3-D immersive and interactive visualization system.

15. SUBJECT TERMS

immersive visualization, 3-D interactive visualization, scientific visualization, virtual reality, real-time ray tracing

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Simon Su
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UU	20	**19b. TELEPHONE NUMBER (Include area code)** (410) 278-1421

Contents

List of Figures

Approved for public release; distribution is unlimited.

1. Introduction

Virtual reality (VR)-enabled scientific visualization workflows can be a useful scientific tool for scientists and stakeholders to gain a better understanding of data during their data exploration and discovery process.[1–7] However, 3-D immersive and interactive visualization capability alone is not a workflow that is sufficient for scientists to employ in the real world. Other than user-friendly software and hardware setup, scientists also need to be able to perform their usual data analysis task in the workflow.

The following are some factors considered by scientists when selecting a visualization workflow:

- Functionality of the software
- Performance of the software
- Ease of hardware setup
- Ease of software setup

Scientists have their favorite data analysis and data visualization tools that they are comfortable using in their day-to-day workflow. For existing data, it is very likely that a desktop version of the visualization software is also available. Desktop visualization tools have worked their way into scientist's day-to-day workflows since desktop computers became ubiquitous and increasingly capable of handling larger data. In the beginning of the year 2016, we saw an explosion in the availability of VR-enabled hardware devices. There is also a surge in efforts by the application development community to develop more VR-enabled applications. As 3-D immersive and interactive visualization can also be a useful research tool, the scientific visualization community has been working on developing a universal 3-D immersive and interactive scientific visualization application.

Although there are overlapping research goals, the VR and scientific visualization communities mostly have different research priorities. For the VR community, the ability to support real-time user interaction is very important. However, for the scientific visualization community, high-resolution visualization takes higher priority. Unfortunately, as real-time interaction and high-resolution visualizations are competing for the same hardware resources, VR and scientific visualization researchers have been forced to concentrate on their own respective research priorities.

To have a VR-enabled scientific visualization capability for their existing workflows, the research community will need to add 3-D immersive and interactive visualization capabilities into their existing desktop visualization applications to inherit the full set of desktop data analysis capabilities for their VR-enabled visualization applications. Shetty et al. added Immersive ParaView as a plugin to ParaView to enable 3-D immersive and interactive visualization using ParaView on some of the popular VR hardware.[8] However, the ParaView rendering pipeline was not optimized for real-time rendering, which has limited the visualization using Immersive ParaView to smaller data sets. Su et al. discovered the same limitation with 3-D immersive and interactive visualization using EnSight.[9] Similarly, real-time rendering performance limitations were experienced with some of the other commercially available VR-enabled visualization packages, including Avizo 3-D.

An alternative method is to reimplement all of the data analysis features of the desktop visualization application in a visualization application developed for the VR platform. Using the Unity3-D game engine's VR plugins for zSpace, researchers developed a VR-enabled scientific visualization prototype application to visualize high-resolution 3-D simulation data. However, visualization of the simulation data at resolutions capable of showing interesting data characteristics has also reduced the interactive VR application to a slow slide show. In addition to the slow rendering performance, reimplementing the desktop data analysis capabilities in VR-enabled visualization software has proven to be a challenge. Kreylos et al. developed, from scratch, a VR-enabled LidarViewer application with all of the data analysis capabilities required.[4] However, these capabilities were still a subset of all of the LiDAR data manipulation features available in similar desktop-based LiDAR data applications.

Lacking the real-time rendering performance and the full data analysis capability of the desktop version has severely limited the adaptation of the VR-enabled visualization workflow. Unwilling to put extra effort into VR-enabled visualization workflows, scientists are often forced to accept the reduced visualization capability of the desktop version of the visualization software. Ease of use with a full set of data analysis capability has always been a higher priority.

Recent advancements in VR hardware have lowered costs and significantly improved reliability. The development of zSpace, a semi-immersive desktop VR device, has made user friendly VR hardware available for scientific visualization. The advances in software technology, specifically the effort to add real-time rendering capabilities in ParaView, has resulted in rendering performance that surpasses the minimum requirements for VR. Therefore, in addition to having a full set of data analysis capability, ParaView is also capable of interactive rendering.

Three-dimensional scientific visualization is a complex problem where visualizing the data in 3-D alone may not provide the user with all necessary insights needed to understand the data. Some 2-D visualization techniques on 3-D data can provide the users with additional insight crucial to the overall understanding of the data. Therefore, a hybrid of 2-D and 3-D visualization can be beneficial to the data exploration and discovery process. Depending on how zSpace is set up, an additional 2-D monitor can easily be added to augment the 3-D visualization capabilities of zSpace with 2-D visualization. Furthermore, ParaView—which was originally designed for 2-D visualization with 3-D immersive visualization added—has the ability to take advantage of both 2-D and 3-D visualization devices at the same time. With the combination of all of these technologies (i.e., ParaView + pvOSPRay + Immersive ParaView + zSpace), we believe that we have the correct combination of hardware and software technologies to offer users a fully functional and user friendly 2-D and 3-D hybrid VR-capable scientific visualization workflow as shown in Fig. 1.

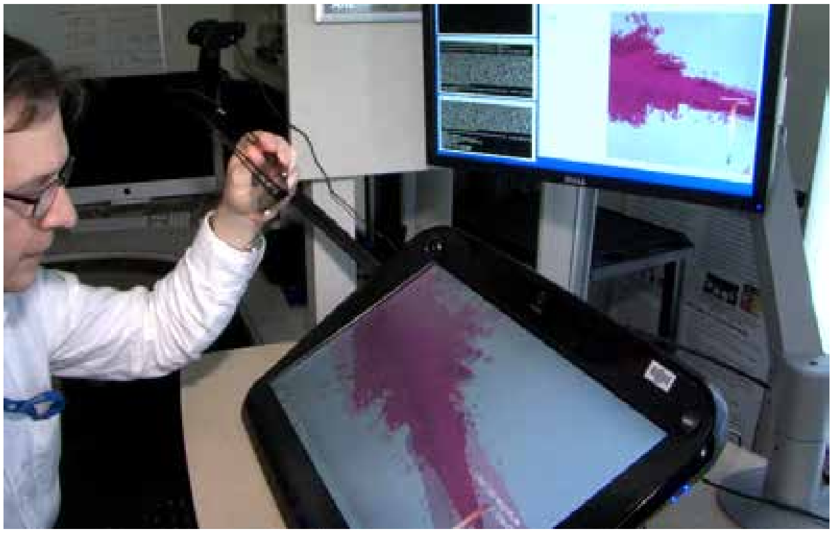

Fig. 1 Immersive ParaView on zSpace

Section 2 discusses the related technologies that we used in our work. We then describe our proposed workflow and our enhancement to the existing pvOSPRay rendering plugin for ParaView. A preliminary experimental study design was discussed together with the result of our user study. Section 2 presents our plan to deploy the technology to our users.

2. Related Work

Researchers have developed various VR-enabled scientific visualization applications and tools that have proven useful to the scientific community. Knabb et al. used VR-enabled 3-D immersive visualization to support their archeological research. Gerndt et al. used immersive visualization technology to support the visualization of their costal restoration modeling.[2] Kreylos et al. developed a fully immersive visualization tool for geoscience that allows a life-size walkthrough of geoscience data.[4] This application is uniquely valuable to the geoscience community because it allows geoscientists to be fully immersed in the data collected from geographic areas with limited access. The data analysis capability of the system allows geoscientists to perform valuable measurements on the data visualized in a fully immersive environment. Sherman et al. used VR technology to support fully immersive visualization of wild forest fire.[5] Su et al. used VR technology to create immersive visualization of geoscience data and hydrological research data where both simulation data and sensor data were used to create fully immersive scientific visualization experiences.[6,7]

Recent enhancements that improve real-time rendering capability in ParaView were leveraged heavily in our workflow. Our workflow was designed to allow the normal use of ParaView to include 3-D immersive and interactive visualization when necessary.

2.1 Immersive ParaView

Immersive ParaView extends ParaView to include VR functionalities by adding multiple-display support and provides a mechanism for ParaView to interface with VR tracking systems. Immersive ParaView supports both Vrui and Virtual Reality Peripheral Network (VRPN) VR devices' communication protocols. ParaView uses the 3-D tracking data from the server to update both the eye-transform matrix and model-transform matrix. Head-tracking data is used in the off-axis projections calculation to support a head-tracked view. The use of 3-D tracking data together with multiple-display support, allows ParaView to be configured to run on a variety of VR systems.[8]

2.2 OSPRay

Intel's OSPRay[*] ray tracing framework provides an optimized central processing unit (CPU) ray tracing library with proven performance capable of rendering large data sets interactively with advanced shading effects. OSPRay was designed from

[*] A ray tracing-based rendering engine for high-fidelity visualization.

the ground up as an underlying rendering library and application programming interface (API) that could be implemented in a wide range of applications across many domains. It currently has implementations in Visualization Toolkit (VTK), ParaView, VisIt, and Visual Molecular Dynamics (VMD)—among other visualization tools.[10] OSPRay's use of CPU rendering means that our implementation can use expansive system memory, allowing for the storage of large-scale data that are high resolution or time varying while still fitting in memory on a single workstation. OSPRay has been shown rendering over 400 GB of data interactively with ambient occlusion at a resolution of 4 MP while rendering point data as implicit spheres.[11] With stereo rendering at 2 MP, we can expect similar performance given that the image is rendered twice at half the resolution, as ray tracing tends to scale linearly with resolution.

OSPRay uses the Intel Embree[12] ray tracing kernels for fast ray traversals. These low-level ray tracing kernels provide optimized spatial acceleration structures with build times of up to more than 100 triangles-per-second, and fast traversal speeds to compute geometry intersections allowing for real-time rendering. For other rendering work such as ray generation and shading, the Intel SPMD Program Compiler is used—allowing for variable-width vector instruction set architectures.[13] Intel's Threaded Building Blocks library allows for efficient thread utilization for many-core workstations.

OSPRay was built as a plugin for VTK and then integrated into ParaView. In the recently released ParaView 5.1.0, OSPRay was fully integrated directly with VTK and ParaView itself. Our stereoscopic rendering support was built into the pvOSPRay plugin, and we hope to have it fully integrated with the built in pvOSPRay version and included in a future release of ParaView.

2.3 pvOSPRay

pvOSPRay follows a similar implementation to the vtkManta plugin.[14] Specific OSPRay classes override underlying VTK classes such as vtkOSPRayCamera and vtkOSPRayRenderer. These custom objects are instantiated through a custom RenderView, but the ParaView guided user interface (GUI) otherwise functions identically for supported geometric and volumetric representations. Ray tracing specific capabilities—such as ambient occlusion and path tracing—are supported through the GUI as custom widgets.

3. Hybrid 2-D and 3-D Immersive Visualization

Our proposed workflow was designed to offer users the immersive visualization capability without leaving the comfort of their office. As shown in Fig. 1, a typical immersive visualization session starts with the ParaView 2-D interface loaded on the 2-D desktop monitor. To support real-time rendering of larger 3-D data sets, users will have to replace the default renderer in ParaView with the pvOSPRay renderer developed by Intel/TACC collaborations as a rendering plugin to ParaView. Using the client/server connection option in ParaView, the user connects to the rendering server configured with zSpace display parameters. Upon a successful connection, a stereoscopic render window will display on zSpace. Using the Immersive ParaView plugin interface, the user can then set up the connection to the VRPN server[15] to enable head-tracked visualization.

Once the connections to the render server and tracking server are established, users can set up their visualization pipeline in ParaView using the mouse and keyboard. At this point, visualizations rendered on the ParaView GUI will also be rendered in stereoscopic mode on the render window on zSpace. Using zSpace circular polarized stereo glasses equipped with fiducial markers, users are able to experience a head-tracked stereoscopic visualization. Although Immersive ParaView supports 3-D interaction capability in the 3-D view, none of the ParaView pipeline manipulation functionalities are available in the 3-D view. However, users will still be able to use the mouse and keyboard to interact with ParaView GUI running on the 2-D desktop monitor for visualization pipeline manipulations to update the visualization on zSpace.

3.1 Detailed Implementation

Although the different ParaView plugins used in our workflow are fully functional, they were not designed and developed for our specific use case. Therefore, our main contributions are the following:

- Update pvOSPRay rendering plugin to add stereo rendering
- Update pvOSPRay rendering plugin to interface with Immersive ParaView plugin
- Update server side rendering code to support visualization of data at multiple scale

We built our work on ParaView 4.4.0 as the pvOSPRay renderer was originally developed as a rendering plugin for that version of ParaView and the pvOSPRay real-time rendering capability is a crucial component in our workflow. Although

ParaView internally supports stereo rendering in its default renderer, this functionality was not exposed to plugin renderer. Because pvOSPRay renderer was intended to support real-time ray tracing rendering, stereo display options were not part of the initial design. For our stereo implementation, we had to modify vtkOpenGLRenderWindow class to expose the calls for setting and getting the display buffer for both left and right eyes. Then, we made modifications to vtkOSPRayCamera and vtkOSPRayRenderer, which support rendering virtual left and right eyes. The renderer is called twice, and we render twice into separate buffers using an offset camera to represent the eye positions. The ParaView pipeline otherwise remains unchanged and functions normally with stereoscopic ray tracing doing the rendering work.

Furthermore, as the pvOSPRay plugin was also not originally designed to work with the Immersive ParaView plugin, the mechanism to update camera position in pvOSPRay with 3-D tracking data from Immersive ParaView plugin was not implemented. However, this can be done by multiplying the camera position in pvOSPRay renderer by the EyeTranformMatrix, which is set by tracking data from the VRPN server.

From a scientific data visualization point of view, the relative scale of data visualized is not the most important factor since data from a single simulation is mostly generated at the same scale factor. ParaView will automatically display the data in full scale of the rendering window. However, visualization in a virtual environment often has to deal with multiple 3-D objects at different sizes and scales. Although more research is needed, we have decided to not visualize the data on the client side at full scale but rather at the scale written in the data set.

3.2 Hardware Setup

As shown in Fig. 1, we are running the proposed workflow on a Dell T7500 graphics workstation equipped with nVidia Quadro M6000 and 24 G of system memory. Both zSpace 100 and 2-D desktop monitors are attached to the video card using display port cables. The zSpace 100 was set up as the primary monitor with desktop extended to the 2-D desktop monitor.

The zSpace 100 stereoscopic display device is equipped with 2 infrared cameras for fiducial-based head tracking and 2 other cameras to support stylus-based 3-D interaction with tracking volume roughly the size of the display. Although this is a relatively small VR setup when compared to the CAVETM it has good VR performance reliability, which is suitable to support hybrid 2-D and 3-D immersive visualization in an office environment.

4. User Interaction Survey

An informal user interaction survey was given at an internal staff meeting across different centers where 8 staff members with expertise in different areas of visualization participated. Staff members have experience in both the creative and technical sides of scientific visualization and have worked to support high-performance computing users with scientific visualization of their data.

Visualization of data sets with 4 different dispersion characteristics from an engine were shown and users were asked questions about the 2-D and 3-D visualizations. Figure 2 shows the 2-D images used in the study and Fig. 1 shows the 3-D experimental setup with the same data set. Although 4 different data sets were used in the 2-D visualization questionnaires, only one data set was used in the 2-D and 3-D comparative part of the study.

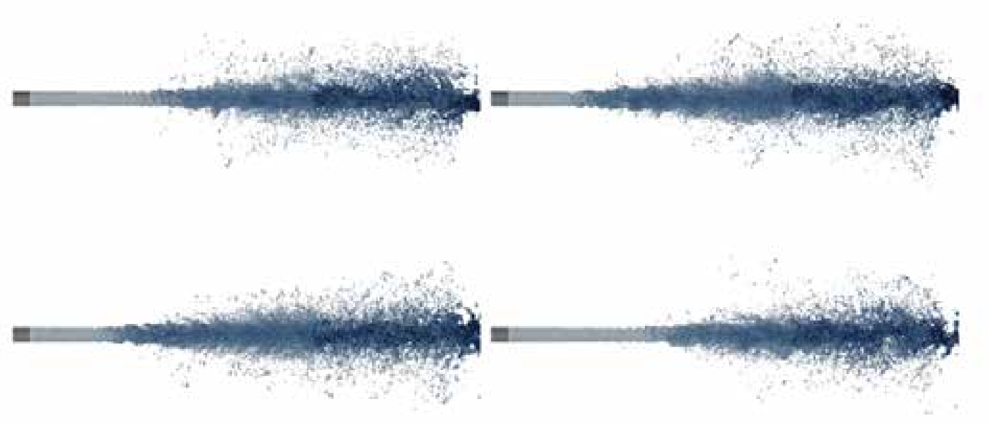

Fig. 2 2-D visualization of 4 different test cases

For the first set of questions, users were counterbalanced such that half saw 2-D first and then 3-D and vice versa for the other half of each of the 4 dispersion visualizations. We asked users (who were all naïve as to the actual data and were not propulsion experts) questions regarding the structure of the data they were viewing. For example, they were asked questions such as, “The greatest amount of droplets over the course of the simulation is for . . .” and then they were asked to pick 1 of the 4 visualizations. After they completed the task in 2-D and 3-D, they were asked comparison questions. For example, “Which mode (i.e., 2-D or 3-D) was easier to determine maximum dispersion?”

Since there was no “correct” answer we aimed to look at the results in terms of a naïve viewer’s preference for display type. Unfortunately, there was no clear “winner” among the dispersion methods. However, 3-D was preferred overall but not by a large margin. Of the 40 responses (8 participants each answering 5 questions), 28 preferred 3-D. The end of the survey was very telling; participants were asked open-ended questions about what they liked and disliked about both displays. For the 2-D display most users said the clarity, resolution, and shading were the best features, while the inability to interact was the worst feature. For the 3-D display the interactivity and depth were preferred, while the lack of resolution and poor shading were lacking. When presented with the choice to only use one or the other, 6 of 8 participants chose 3-D and the other 2 said they would choose 3-D if the resolution and shading could be improved.

5. Discussion and Future Work

Informal user interaction survey findings are encouraging. Demonstrating the workflow to computational scientists exposes interest in the flexibility of our hybrid 2-D and 3-D VR-enabled workflow. Talking to the scientific visualization developer community (i.e., ParaView and VisIt developers) has also generated interest in adopting our workflow.

Our work was based on the ParaView 4.4.0 code base; changes in the latest release of ParaView has complicated the plan to merge our enhancements with ParaView. However, we are working closely with Kitware to add the enhancement to a standard release of ParaView. With OSPRay fully integrated into ParaView and VTK, changes to make stereo rendering may not be necessary. However, we believe we can improve the stereo rendering quality of ParaView. With the latest release of ParaView 5.1.0, we have a stable code base to migrate our enhancements into ParaView.

We are also planning to conduct a more comprehensive user study to determine the usability of our workflow. Our main interest with the proposed hybrid workflow is to support a VR-enabled universal scientific visualization application. Working toward that goal, if our usability study using nonexpert participants is showing a positive outcome, we plan to also support EnSight and VisIt visualization toolkits in our workflow.

6. Conclusion

We developed and tested a hybrid 2-D and 3-D VR-enabled scientific visualization workflow. The workflow allows users to retain their existing interactions with ParaView in their office space with 3-D immersive and interactive visualization capabilities. The hybrid 2-D and 3-D user interface is providing the user with complimentary insight into their data exploration and discovery process.

The zSpace VR device is the only additional hardware needed to enable 3-D immersive and 3-D interactive visualization capabilities. Although, the Immersive ParaView plugin will also be needed to run ParaView on zSpace, the user experience running ParaView is unchanged. The new native OSPRay renderer support in ParaView also improves ease of use. Comments by visualization experts from our informal user study indicated a preference for a higher resolution display. However, a different VR hardware setup will be needed to support higher display resolutions.

7. References

1. Gallus WA, Cervato C, Cruz-Neira C, Faidley G, Heer R. A virtual tornadic thunderstorm enabling students to construct knowledge about storm dynamics through data collection and analysis. 13th Symposium on Education; 2009 Jan 11–15; Seattle, WA.

2. Gerndt A, Miller R, Meselhe E, Cruz-Neira C, Su S. Interactive exploration of coastal restoration modeling in virtual environment. IS&T/SPIE Electronic Imaging Science and Technology 2009; 2009 Jan. p. 7238–15. Paper.

3. Knabb KA, Schulze JP, Kuester F, DeFanti TA, Lev TE. Scientific visualization, 3D immersive virtual reality environments, and archaeology in Jordan and the Near East. Near Eastern Arch. 2014 Sep:77(3);228–232.

4. Kreylos O, Bawden G, Bernardin T, Billen MI, Cowgill ES, Gold RD, Hamann B, Jadamec M, Kellogg LH, Staadt OG, Sumner DY. Enabling scientific workflows in virtual reality. Proceedings of the 2006 ACM International Conference on Virtual Reality Continuum and its Applications.

5. Sherman W, Penick M, Su S, Harris F, Brown T. VRFire: an immersive visualization experience for wildfire spread analysis. IEEE Virtual Reality 2007; 2007 Mar. p 243–246.

6. Su S, Sherman W, Harris F, Dye M. TAVERNS: visualization and manipulation of GIS data in 3D large screen immersive environments. The Proceedings of 16th International Conference on Artificial Reality and Telexistence; 2006 Nov.

7. Su S, Cruz-Neira C, Habib E, Gerndt A. Virtual hydrology observatory: an immersive visualization of hydrology modeling. IS&T/SPIE Electronic Imaging Science and Technology 2009; 2009 Jan. p. 7238–17. Paper.

8. Shetty N, Chaudhary A, Coming D, Sherman WR, O'Leary P, Whiting ET and Su S. Immersive ParaView: a community-based, immersive, universal scientific visualization application. Virtual Reality Conference (VR); 2011; IEEE. p. 239–240.

9. Su S, Chaudhary A, O’Leary P, Geveci B, Sherman W, Nieto H, Francisco-Revilla L. Virtual reality enabled scientific visualization workflow Everyday Virtual Reality (WEVR). 2015 IEEE 1st Workshop; 2015 Mar 23–23. p. 29–32. 10.1109/WEVR.2015.7151692.

10. Wald I, Brownlee C, Amstutz J, Knoll A, Guenther J. OSPRAY- a CPU ray tracing framework for scientific visualization. IEEE Trans on Vis and Comp Gfx. 2017 Jan;23(1):931-940. DOI: 10.1109/TVCG.2016.2599041.

11. Wald I, Knoll A, Johnson GP, Usher W, Pascucci V, Papka ME. CPU ray tracing large particle data with balanced P-k-d trees. 2015 IEEE Scientific Visualization Conference (SciVis); 2015 Oct 25–30. p. 57–64. 10.1109/SciVis.2015.7429492.

12. Wald I, Woop S, Benthin C, Johnson GS, Ernst M. Embree: a kernel framework for efficient CPU ray tracing. ACM Trans on Gfx. 2014 Jul; 33(4). doi>10.1145/2601097.2601199.

13. Pharr M, Mark B. ISPC: A SPMD compiler for high-performance CPU programming. In Proceedings of Innovative Parallel Computing (inPar); 2012. p. 184–196.

14. Brownlee C, Patchett J, Lo LT, DeMarle D, Mitchell C, Ahrens J, Hansen C. A study of ray tracing large-scale scientific data in two widely used parallel visualization applications. Eurographics Symposium on Parallel Graphics and Visualization (EGPGV âA˘Z´12); 2012. p. 51–60.

15. Taylor RM 2nd, Hudson TC, Seeger A, Weber H, Juliano J, Helser AT. VRPN: a device-independent, network-transparent VR peripheral system. VRST ’01: Proc of the ACM symposium on VR software and technology; 2001; New York. New York (NY): ACM; c2001. p. 55–61.

List of Symbols, Abbreviations, and Acronyms

2-D	2-dimensional
3-D	3-dimensional
API	application programming interface
CPU	central processing unit
GUI	guided user interface
VMD	Visual Molecular Dynamics
VR	virtual reality
VRPN	Virtual Reality Peripheral Network
VTK	Visualization Toolkit

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIR ARL
(PDF) RDRL CIO L
IMAL HRA MAIL & RECORDS
MGMT

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

2 ARL
(PDF) RDRL CIH S
S SU
RDRL VTP
L BRAVO